2階導関数を使うRunge-Kutta型公式の探索 (数式処理と数学研究への応用)

三井, 斌友

数理解析研究所講究録
406巻 1980年 116-134

2階導関数を使うRUNGE-KUTTA型公式の探索

京大　数理研　　三井斌友

0. はじめに

筆者は数式処理Symbolic and Algebraic Manipulation (SAM) システムを専門にしているものではないが，常微分方程式の初期値問題の数値解析の研究にSAMを用いた，userとしての経験を報告させていただくのが，本稿の目的である．数理研では，DEC System 2020 にREDUCE-2 を導入しているが，そのすべての機能を用いたのではなく，はなはだ貧しい経験であるが，幸い問題が丁度SAMにのせやすいsizeであることもあって，有効に利用できている．

1. p-stage RUNGE-KUTTA公式

微分方程式の初期値問題

(1.1)　$\dfrac{dy}{dx} = f(x, y)$, $x \geqq x_0$,　　(1.2)　$y(x_0) = y_0$

を数値積分するone-step methodとして，まずTaylor展開を用いることが考えられる．step-sizeをhとして，(1.1)(1.2)の

解 $y(x)$ の x_0+h での値の近似値を

(1.3) $y_1 = y(x_0) + hf(x_0, y(x_0)) + \frac{1}{2}h^2 f'(x_0, y(x_0)) + \cdots + \frac{1}{n!}h^n f^{(n-1)}(x_0, y(x_0))$

によって定める．$y(x_0+h)$ の Taylor 展開との比較で分るように、この方法の局所打切り誤差 local truncation error は $O(h^{n+1})$ であるから，方法は order n と呼ばれる．

(1.3) に現われる導関数 f', f'', $\cdots$, $f^{(n-1)}$ はそれぞれ

$\frac{d}{dx}f(x, y(x))$, $(\frac{d}{dx})^2 f(x, y(x))$, $\cdots$, $(\frac{d}{dx})^{n-1} f(x, y(x))$

の意味で，微分方程式 (1.1) が具体的に与えられれば（すなわち $f(x, y)$ の関数形が与えられれば），chain rule によって計算できる．しかし，高階導関数を実際に求めることは，それ自身手間をとるので，Taylor 法は実用性に乏しいとされている．

（注）$f(x, y)$ の関数形を与えて，その導関数を関数形で求めることも，SAM で可能なことである．事実，SAM システムの発達は，Taylor 法に対する上のような主張を解消する可能性があると指摘されている [5]．

そこで，解関数の1階導関数である $f(x, y)$ の evaluation を何度か行ない，それらを組み合わせて，Taylor 法と同じ order の accuracy をもつ方法が考えられた．これを総称して RUNGE-KUTTA 法と呼んでいる．one-step の間に行なう，$f(x, y)$ の evaluation の回数を stage 数というが，p-stage RK 公式は，次のように表わされる．

$$(1.4)\quad \begin{cases} y_1 = y_0 + h \sum_{i=1}^{p} a_i k_i \\ k_i = f(x_0 + c_i h,\ y_0 + h \sum_j b_{ij} k_j) \end{cases}$$

ここで，$b_{ij} = 0$ for $j \geq i$ なら explicit type，$b_{ij} = 0$ for $j > i$ なら semi-explicit type，そうでないとき implicit type と呼ぶ．

ふつう使われているRK公式（classical RK公式）は，$p=4$ の explicit type である．

RK公式を決定するには，k_i を二変数関数として Taylor 展開し，y_1 を h のべきで整理して，$y(x_0+h)$ の Taylor 展開と比較し，必要な order まで h のべきの係数が一致するようにして得られるパラメータ a_i，b_{ij}，c_i の関係式を解くことが必要である．これは全く代数的計算であるが，order が上っていくと大変な量になる．これを非常にみとおしよく整理したのが J. C. BUTCHER の一連の研究（[1]〜[4]）で，RK公式に対して画期的展望を与えた．その結果，当時未解決であった 5-stage 公式の order はいくつにできるかという問題に，order 4 という解答を与えたばかりでなく，10-stage までの公式の attainable order を決定した．さらに implicit type や semi-explicit type を新しい概念として導入し，それらが公式の stability について新しい展望を与えた．

BUTCHER の研究では，実用上は無理がある（linear multi-step method である PC 法などと比べて）と思われる implicit typeが積極的に用いられていることに注意したい．

2. 2階導関数を使うRUNGE-KUTTA公式

RK法が1階導関数$f(x,y)$のevaluationに限定していたのは自然であるが，少し立場を変えて，2階導関数

$$g(x,y) \equiv f_x(x,y) + f_y(x,y) f(x,y)$$

のevaluationも許すとすると，one-step methodsの新しいseriesができる．

（注）$g(x,y)$の具体的な形は，次の例でみていただきたい．

$$f(x,y) = \begin{bmatrix} f_1(y_1, y_2) \\ f_2(y_1, y_2) \end{bmatrix} = \begin{bmatrix} (a_1 - b_1 y_2) y_1 \\ (a_2 - b_2 y_1) y_2 \end{bmatrix} \quad (a_i, b_i : \text{constants})$$

ならば

$$g(x,y) = \begin{bmatrix} a_1^2 y_1 + (b_2 y_1 + b_1 y_2 - 2a_1 - a_2) b_1 y_1 y_2 \\ a_2^2 y_2 + (b_2 y_1 + b_1 y_2 - a_1 - 2a_2) b_2 y_1 y_2 \end{bmatrix}.$$

2階導関数$g(x,y)$のevaluationを含む公式を初めて作ったのは占部[11]のようである．その後新谷[7, 8]，CASH [6]などがみられる．占部，CASHは線形多段法に，新谷はRK法にanalogausとみることができる．

われわれは，$g(x,y)$を含む一般のRK型公式，(p,q)-stage公式を次のように定式化する．

$$(2.1)\begin{cases} y_1 = y_0 + h\sum_{i=1}^{p}\mu_i k_i + h^2\sum_{i=1}^{q}\nu_i K_i, \\ k_i = f(x_0+\alpha_i h,\ y_0 + h\sum \beta_{ij}k_j + h^2\sum_j \gamma_{ij}K_j), \\ K_i = g(x_0+\rho_i h,\ y_0 + h\sum_j \sigma_{ij}k_j + h^2\sum_j \tau_{ij}K_j). \end{cases}$$

この公式の理論的可能性として，その attainable order を定めていこう．理論的研究のためには implicit type の方が都合がよいので，(2.1) は implicit としておく．$(p,0)$-stage formula は，ふつうの意味の RK 公式 (1.4) になるから，(p,q)-stage formula はひとつの一般化となっている．

（注）戸田 [10] は，RK 公式の 5 次の極限公式を扱っている（本講究録中の戸田・小野両氏の論文も同趣旨）．そこでは 5-stage の公式で，任意に決められるパラメータをある極限値に移行すると，$k_i - k_{i+1}$ のある値は 0 に近くなることが示されている．もし，それをそのまま採用すれば 2 階導関数を含むわれわれの公式に近いものになると思われる．[10] では RK 法の通常の約束で，$f(x,y)$ の evaluation だけを行なうことにしているので，$g(x,y)$ は現われない．

われわれの RK 公式の研究のためには，まず $y(x_0+h)$ の h のべき級数への展開を，g とその偏導関数を用いて表わしておくと好都合である．そのために，次の定理を示すことができる．

<u>Theorem 1.</u> $y(x_0+h)$が，hのべき級数として

(2.2) $$y(x_0+h)=y_0+hf_0+\sum_{r=2}^{m}\frac{\kappa_{r-2}}{r!}h^r+O(h^{m+1})$$

と展開されるとしよう．すると，κ_lは

(2.3) $$\begin{cases}\kappa_0=g_0\\ \kappa_l=[D_0^l g]_0+\sum_{s=1}^{l-1}\left\{\sum_{t=1}^{m(l,s)}\frac{l!}{s!\,(l-s-t)!}B_{s,t}\left(\frac{\kappa_0}{2},\cdots,\frac{\kappa_{s-1}}{s+1}\right)\left[D_0^{l-s-t}\left(\frac{\partial}{\partial y}\right)^t g\right]_0\right\},\end{cases}$$

$$l=1,2,\cdots,m-2$$

と与えられる．ここでsuffix 0は$x=x_0$におけるevaluationを表わし，$D_0=\frac{\partial}{\partial x}+f_0\frac{\partial}{\partial y}$，$m(l,s)=\min(s,l-s)$，$B_{s,t}$は$(s,t)$次のBELL多項式である．

(注) 微分方程式(1.1)が連立である場合，f_yはJacobian行列であって，gは$f_x+f_y\cdot f$の順序で書かれるのが正しい．しかし公式のorderの研究のためには，単独方程式で考えれば十分であるので，yに関する偏導関数とfやgとの積は，順序を問わない．

(2.3)式において，κ_lを決めるには，右辺で必要なのはκ_{l-2}までであることに注意すると，これはSAMシステムにかけられるalgorithmになっている．ただし，BELL多項式が求められている必要があるが，BELL多項式もそれを生成するalgorithmが知られている．すなわち

$$\sum_{t=1}^{s}x_1B_{s,t}(x_1,\cdots,x_s)a^{t+1}+\sum_{t=1}^{s}\left(\sum_{k=1}^{s}x_{k+1}\frac{\partial B_{s,t}}{\partial x_k}\right)a^t$$
$$=\sum_{t=1}^{s+1}B_{s+1,t}(x_1,\cdots,x_{s+1})a^t.$$

ここで両辺はaの多項式として恒等式とする.

また(2.3)式から, κ_ℓは, $[D_0^s(\frac{\partial}{\partial y})^t g]_0$ (s, tは非負整数)の形の項のいくつかの積の1次結合(係数は実は正整数になる)である. そこで$\prod_{s,t}[D_0^s(\frac{\partial}{\partial y})^t g]_0$のことをelementary differentialsと呼ぶことにする.

筆者はREDUCE-2によって, BELL多項式を15次まで, (2.3)によってκ_ℓを$\ell=8$まで, それぞれおめた. 特に各κ_ℓをおめて, そこに含まれているelementary differentialsの数を調べておくと, 役に立つ.

なお, Th. 1の考え方は, $y(x_0+h)$をfとその偏導関数を用いてhのべき級数として表わすことにも応用できる.

<u>Theorem 2.</u> $y(x_0+h)$がhのべき級数として

$$(2.4)\quad y(x_0+h)=y_0+\sum_{r=1}^{m}\frac{\lambda_{r-1}}{r!}h^r+O(h^{m+1})$$

と展開されるとする. すると, λ_ℓは

$$(2.5)\begin{cases}\lambda_0=f_0,\\ \lambda_\ell=[D_0^\ell f]_0+\sum_{s=1}^{\ell-1}\{\sum_{t=1}^{m(\ell,s)}\frac{\ell!}{s!(\ell-s-t)!}B_{s,t}(\frac{\lambda_1}{2},\frac{\lambda_2}{3},\cdots,\frac{\lambda_s}{s+1})[D_0^{\ell-s-t}(\frac{\partial}{\partial y})^t f]_0\},\\ \qquad \ell=1,2,\cdots,m-1\end{cases}$$

によって与えられる.

BUTCHERは, われわれのとは違った意味でelementary differentialsを定義して, $y(x_0+h)$のhのべき級数展開の係数をそれによって表わしたが, Th. 2はBUTCHERの結果の別の形の表現

となっている.

（注）(2.2) と (2.4) は $y(x_0+h)$ の h のべき級数としての二通りの表現である. そして実際, g に関する elementary differentials を構成している $[D_0^s (\frac{\partial}{\partial y})^t g]_0$ に対しては,

$$[D_0^s g]_0 = [D_0^{s+1} f]_0 + \sum_{r=0}^{s-1} \binom{s}{r} [D_0^{s-r} f]_0 \cdot [D_0^r f_y]_0$$

$$[D_0^s g_y]_0 = [D_0^{s+1} f_y]_0 + \sum_{r=0}^{s} \binom{s}{r} [D_0^{s-r} f_y]_0 [D_0^r f_y]_0 + \sum_{r=0}^{s-1} \binom{s}{r} [D_0^{s-r} f]_0 \cdot [D_0^s f_{yy}]_0$$

$$\vdots$$

の恒等式がなりたつから, K_ℓ は, 上の substitution を行なうことによって, f に関する elementary differentials を使った表現に変換できる. 筆者は, DEC 2020 の software FILCOM（二つの file の内容の比較）を用いて, λ_ℓ と, 変換された K_ℓ とを比較し, 同一であることを確認できた.

3. (1, g)-stage formula

一般の (p, g)-stage formula を扱うことは相当困難なので, 次のような特別の形の implicit $(1, g)$-stage formula をまず考察しよう.

$$\begin{cases} y_1 = y_0 + \mu_1 h k_1 + \sum_{i=1}^{g} \nu_i h^2 K_i, \\ k_1 = f(x_0 + \alpha_1 h,\ y_0) \\ K_i = g(x_0 + \rho_i h,\ y_0 + \sigma_{i1} h k_1 + \sum_j \tau_{ij} h^2 K_j),\quad i = 1, \cdots, g \end{cases}$$

$\alpha_1 = 0$, $\mu_1 = 1$, $\sigma_{i1} = \rho_i$ であることがすぐ分るので, 実は公式

は

$$(3.1)\quad \begin{cases} y_1 = y_0 + hf_0 + \sum_{i=1}^{g} \nu_i h^2 K_i, \\ K_i = g(x_0 + \rho_i h,\ y_0 + \rho_i h f_0 + \sum_j \tau_{ij} h^2 K_j), \quad i=1, \cdots, g \end{cases}$$

である．これは，explicit type の場合，新谷[7]と一致する．

パラメータ ν_i, ρ_i, τ_{ij} を決定する条件を導かねばならない．Th. 1 と同じ考え方で，次のことがいえる．

<u>Theorem 3.</u> 各 i について，K_i を h のべき級数として

$$(3.2)\quad K_i = \sum_{\ell=0}^{m-2} \frac{\kappa_{i\ell}}{\ell!} h^\ell + O(h^{m-1})$$

と表わす．すると $\kappa_{i\ell}$ は

$$(3.3)\quad \begin{cases} \kappa_{i0} = g_0, \\ \kappa_{i\ell} = \rho_i^\ell [D_0^\ell g]_0 + \sum_{s=1}^{\ell-1} \Big\{ \sum_{t=1}^{m(\ell,s)} \frac{\ell!}{s!(\ell-s-t)!} \rho_i^{\ell-s-t} B_{s,t}(\sum_j \tau_{ij}\kappa_{j0}, \cdots, \\ \qquad s\sum_j \tau_{ij}\kappa_{j,s-1}) [D_0^{\ell-s-t} (\frac{\partial}{\partial y})^t g]_0 \Big\}, \quad \ell = 1, 2, \cdots, m-2. \end{cases}$$

で与えられる．

Th. 1 の結果とあわせると，各 ℓ に対して

$$(3.4)\quad (\ell+1)(\ell+2) \sum_{i=1}^{g} \nu_i \kappa_{i\ell} = \kappa_\ell, \quad \ell = 0, 1, \cdots, m-2$$

がなりたつ．そして (2.3) と (3.3) を比較すれば，κ_ℓ と $\kappa_{i\ell}$ のなかに含まれる elementary differentials は同一のものとなることが分るから，パラメータに関する条件はすべて

（正整数）×（ν_i, ρ_i, τ_{ij} の多項式）＝（正整数）

となる．すなわちパラメータを決定する方程式の個数は，κ_0 から κ_ℓ までに含まれる elementary differentials の数に等しく，

これと $(1,q)$-stage formulaで決定すべきパラメータの個数との関係を考察すれば，少なくとも到達できるorder，すなわちattainable orderの下限は決まる．

$m(\ell)$を，x_ℓが含んでいるelementary differentialsの数とし，$M(\ell)$を

$$M(\ell)=\sum_{j=0}^{\ell} m(j)$$

と定義する．表1では，$m(\ell)$，$M(\ell)$を $\ell\leq 8$ まであげてある．またimplicit $(1,q)$-stage formulaにおいて決定すべきパラメータの数を，qの関数として$N_I(q)$で表わすと

$$N_I(q)=q(q+2)$$

である．或るqに対して，$M(\ell)\leq N_I(q)$をみたす最大の整数を$\bar{\ell}$とすれば，公式は少なくともorder $(\bar{\ell}+2)$とすることができる．同様にexplicit $(1,q)$-stage formulaにおいて決定すべきパラメータの数を$N_E(q)$とすると

$$N_E(q)=q(q+3)/2$$

である．上と同様にℓ^*を，$M(\ell)\leq N_E(q)$から決定すると，このformulaは少なくともorder (ℓ^*+2)とすることができる．これらの関係は表2にあげてある．

$\bar{\ell}$やℓ^*は，attainable orderの下限に関係するもので，上限の方を与えるには，(3.4)からえられる決定方程式を実際に解いてみなければならない．

Tab. 1

ℓ	$m(\ell)$	$M(\ell)$
0	1	1
1	1	2
2	2	4
3	3	7
4	6	13
5	9	22
6	17	39
7	26	65
8	46	111

Tab. 2

q	$N_I(q)$	$\bar{\ell}$	$N_E(q)$	ℓ^*
1	3	1	2	1
2	8	3	5	2
3	15	4	9	3
4	24	5	14	4
5	35	5	20	5
6	48	6	27	5
7	63	6	35	6
8	80	7	44	6
9	99	7	54	6
10	120	8	65	7

4. Implementation

具体的に決定方程式を書き下すためには，REDUCE-2によって(3.3)を実行させるのであるが，そのためのprogramは次のようになる.

(1) 準備 1. BELL多項式のinput.（別のfileで用意してもよい.）

(2) 準備 2. $\kappa_{i\ell}$, $[D_0^s(\frac{\partial}{\partial y})^t g]_0$, ρ_i, ν_i, τ_{ij} のarrayの用意.

$[D_0^s(\frac{\partial}{\partial y})^t g]_0$ は，DGY(I, J)というarrayを用意し，その各々の値を DIG$\underbrace{\text{Y}\cdots\text{Y}}_{\text{J個}}$ のようにassignしてやることにした.

(3) 準備 3. factorial の計算procedure.

(4) κ_{i0} の計算 (assignment).

(5) BELL多項式の各変数へ，前の結果の代入.

(6) (3.3)による代数計算.

(7) (5), (6)の反復.

(8) 結果を，別のfileにoutput.

programの実際は，本稿の最後に添えてあるので参照していただきたい.

5. むすび

SAMシステム，ここではREDUCE-2は，解析にとって強力な道具である．少なくともこれがなければ，われわれのRK型公式を扱うという気もおこらなかったであろう．更に，パラメータの決定方程式を解く（解けるかどうかの検討も含めて）となると，terminalを傍におきながら，trial and errorを重ねるしか，方法はみつからない．一般的に数学研究にもっともっとSAMシステムを応用すべきであると思う．筆者もなるべく宣伝に努めているが，SAMの専門家の方々にお願いすることとして，二・三のことをあげると

(1) SAMシステムは，interactiveであるべきことは当然だが，fileとの間でinput, outputができることが是非要る.

(2) manualをfamiliarなものにすること．文法書を読んでも良い文章ができるとは限らないと同様，SAMのmanualもtutorialなもの，exampleの多いものが良さそうである．勿論，一旦使いこんだあとではreference manualも要る．やがては数学教科書のstyleもそういう方向に変っていくように，

まず先鞭をつけられたらと思う.

(3) language を使いやすくすること. 私たちのように数値解析に従事し, FORTRAN などによる programming の経験のあるものは, SAM language の理解も早いと思うが, そうでない人達には, やはりその修得には hesitating なようである.

更に, ある部分は SAM, その結果を用いた別の部分は数値処理 (FORTRAN など) という program を可能にする higher level language ができればと期待される.

References

[1] Butcher, J.C., Coefficients for the study of Runge-Kutta integration processes, J. Austral. Math. Soc., 3(1963), 185-201.

[2] ——————, On Runge-Kutta processes of high order, J. Austral. Math. Soc., 4(1964), 179-194.

[3] ——————, Implicit Runge-Kutta processes, Math. Comput., 18(1964), 50-64.

[4] ——————, On the attainable order of Runge-Kutta methods, Math. Comput., 19(1965), 408-417.

[5] Carnahan, B. et al., Applied Numerical Methods, John Wiley, New York, 1969. 邦訳, FORTRAN による数値計算法, 科学技術出版社, 東京, 近刊.

[6] Cash, J.R., High order methods for the numerical integration of ordinary differential equations, Numer.Math., 30(1978), 385-409.

[7] Shintani, H., On one-step methods utilizing the second derivative, Hiroshima Math. J., 1(1971), 349-372.

[8] Shintani, H., On explicit one-step methods utilizing the second derivative, Hiroshima Math. J., 2(1972), 353-368.

[9] Mitsui, T., Runge-Kutta type integration formulas including the evaluation of the second derivative, I, in preparation.

[10] Toda, H., On the truncation error of a limiting formula of Runge-Kutta methods, Researches of Electrotech. Labor., No. 772, 1977.

[11] Urabe, M., An implicit one-step method of high-order accuracy for the numerical integration of ordinary differential equations, Numer. Math., 15(1970), 151-164.

附録 REDUCE-2 programs

I. BELL多項式の生成.

```
COMMENT  THIS IS A REDUCE-2 FILE TO CALCULATE THE BELL'S POLYNOMIAL;

ARRAY B(15,15),X(15),KAP(15);
FACTOR A;

X(1):=X1; X(2):=X2; X(3):=X3; X(4):=X4; X(5):=X5; X(6):=X6; X(7):=X7;
X(8):=X8; X(9):=X9; X(10):=X10; X(11):=X11; X(12):=X12; X(13):=X13;
X(14):=X14; X(15):=X15;

B(1,1):=X1;

NMAX:=14;
FOR N:=1:NMAX DO
  BEGIN K:=0; NP1:=N+1;
    FOR I:=1:N DO K:=K+X(1)*B(N,I)*A**(I+1)+(FOR J:=1:N
                        SUM(X(J+1)*DF(B(N,I),X(J))))*A**I;

    WRITE "(",N,") ",K;
    COEFF(K,A,KAP);
    FOR I:=1:NP1 DO B(NP1,I):=KAP(I);
END;

COMMENT OUTPUT THE RESULTS;

OFF ECHO; OFF ALLFAC; OUT BELLOT.RED;
WRITE "COMMENT  THIS IS A REDUCE-2 FILE TO STORE THE BELL'S ";
WRITE "            POLYNOMIAL AS THE RESULTS RUNNING THE FILE BELPLY.RED;";
WRITE "ARRAY BELL(15,15);";
OFF NAT;
FOR N:=1:NMAX+1 DO
   FOR I:=1:N DO WRITE "BELL(",N,",",I,"):=",B(N,I);
SHUT BELLOT.RED;
```

II. BELL多項式（前のprogramの結果）

```
COMMENT  THIS IS A REDUCE-2 FILE TO STORE THE BELL'S
         POLYNOMIAL AS THE RESULTS RUNNING THE FILE BELPLY.RED;
ARRAY BELL(15,15);
BELL(1,1):=X1$
BELL(2,1):=X2$
BELL(2,2):=X1**2$
BELL(3,1):=X3$
BELL(3,2):=3*X1*X2$
BELL(3,3):=X1**3$
BELL(4,1):=X4$
BELL(4,2):=4*X1*X3 + 3*X2**2$
BELL(4,3):=6*X1**2*X2$
BELL(4,4):=X1**4$
BELL(5,1):=X5$
BELL(5,2):=5*X1*X4 + 10*X2*X3$
BELL(5,3):=10*X1**2*X3 + 15*X1*X2**2$
BELL(5,4):=10*X1**3*X2$
BELL(5,5):=X1**5$
BELL(6,1):=X6$
BELL(6,2):=6*X1*X5 + 15*X2*X4 + 10*X3**2$
BELL(6,3):=15*X1**2*X4 + 60*X1*X2*X3 + 15*X2**3$
BELL(6,4):=20*X1**3*X3 + 45*X1**2*X2**2$
BELL(6,5):=15*X1**4*X2$
BELL(6,6):=X1**6$
BELL(7,1):=X7$
BELL(7,2):=7*X1*X6 + 21*X2*X5 + 35*X3*X4$
BELL(7,3):=21*X1**2*X5 + 105*X1*X2*X4 + 70*X1*X3**2 + 105*X2**2*X3$
BELL(7,4):=35*X1**3*X4 + 210*X1**2*X2*X3 + 105*X1*X2**3$
BELL(7,5):=35*X1**4*X3 + 105*X1**3*X2**2$
BELL(7,6):=21*X1**5*X2$
    ⋮
```

（以下続く）

III. $f(x_0+h)$ の h のべき級数への展開（2.3）

```
COMMENT   REDUCE-2 FILE TO PREPARATE TO GENERATE THE TAYLOR SERIES
          EXPANSION OF SECOND TYPE;

ARRAY BELL(15,15);

BELL(1,1):=X1$

BELL(2,1):=X2$

BELL(2,2):=X1**2$

BELL(3,1):=X3$
```

⋮ (BELL多項式の input)

```
BELL(8,7):=28*X1**6*X2$

BELL(8,8):=X1**8$


ARRAY DGY(8,7),KAPPA(8); OFF ECHO;

DGY(0,0):=G;
DGY(0,1):=GY;
DGY(0,2):=GYY;
DGY(0,3):=GYYY;
DGY(0,4):=GYYYY;
DGY(0,5):=GYYYYY;
DGY(0,6):=GYYYYYY;
DGY(0,7):=GYYYYYYY;

DGY(1,0):=DG;
DGY(1,1):=DGY;
DGY(1,2):=DGYY;
DGY(1,3):=DGYYY;
DGY(1,4):=DGYYYY;
DGY(1,5):=DGYYYYY;
DGY(1,6):=DGYYYYYY;
DGY(1,7):=DGYYYYYYY;


DGY(2,0):=D2G;
DGY(2,1):=D2GY;
DGY(2,2):=D2GYY;
DGY(2,3):=D2GYYY;
DGY(2,4):=D2GYYYY;
DGY(2,5):=D2GYYYYY;
DGY(2,6):=D2GYYYYYY;
DGY(2,7):=D2GYYYYYYY;

DGY(3,0):=D3G;
DGY(3,1):=D3GY;
DGY(3,2):=D3GYY;
DGY(3,3):=D3GYYY;
DGY(3,4):=D3GYYYY;
DGY(3,5):=D3GYYYYY;
DGY(3,6):=D3GYYYYYY;
DGY(3,7):=D3GYYYYYYY;
```

⋮ (DGY(I, J) の assignment)

```
COMMENT   INTEGER PROCEDURE FOR FACTORIAL OF AN INTEGER N;
  INTEGER PROCEDURE FAC(N);
        BEGIN INTEGER M;
                M:=1$
           L1:  IF N=0 THEN RETURN M;
                M:=M*N$
                N:=N-1$
                GO TO L1
        END;
```

(続く)

```
FACTOR G,DG,D2G,D3G,D4G,D5G,D6G,D7G,GY,DGY,D2GY,D3GY,D4GY,D5GY,D6GY,
      GYY,DGYY,D2GYY,D3GYY,D4GYY,D5GYY,GYYY,DGYYY,D2GYYY,D3GYYY,D4GYYY,
      GYYYY,DGYYYY,D2GYYYY,D3GYYYY;


COMMENT  GENERATING PROCEDURE;

KAPPA(0):=DGY(0,0);

X1:=KAPPA(0)/2;

LO:=1;
    INTEGER M,N,LOM1,LOMMN,MINI;
         KAPPA(LO):=DGY(LO,0);
         IF LO<=1 THEN GO TO LAB1;
         LOM1:=LO-1;
         FOR M:=1:LOM1 DO
          BEGIN MINI:=LO-M;
                IF M<=LO-M THEN MINI:=M;
                FOR N:=1:MINI DO BEGIN LOMMN:=LO-M-N;
             WRITE "(",M,",",N,") ";
                 KAPPA(LO):=KAPPA(LO)+
                BELL(M,N)*DGY(LOMMN,N)*FAC(LO)/(FAC(M)*FAC(LOMMN)) END;
          END;
  LAB1: WRITE "(",LO,") ",KAPPA(LO);
   X2:=KAPPA(1)/3;

LO:=2;
         KAPPA(LO):=DGY(LO,0);
         IF LO<=1 THEN GO TO LAB2;
         LOM1:=LO-1;
         FOR M:=1:LOM1 DO
          BEGIN MINI:=LO-M;
                IF M<=LO-M THEN MINI:=M;
                FOR N:=1:MINI DO BEGIN LOMMN:=LO-M-N;
             WRITE "(",M,",",N,") ";
                 KAPPA(LO):=KAPPA(LO)+
                BELL(M,N)*DGY(LOMMN,N)*FAC(LO)/(FAC(M)*FAC(LOMMN)) END;
          END;
  LAB2: WRITE "(",LO,") ",KAPPA(LO);
   X3:=KAPPA(2)/4;

LO:=3;
         KAPPA(LO):=DGY(LO,0);
         IF LO<=1 THEN GO TO LAB3;
         LOM1:=LO-1;
         FOR M:=1:LOM1 DO
          BEGIN MINI:=LO-M;
                IF M<=LO-M THEN MINI:=M;
                FOR N:=1:MINI DO BEGIN LOMMN:=LO-M-N;
             WRITE "(",M,",",N,") ";
                 KAPPA(LO):=KAPPA(LO)+
                BELL(M,N)*DGY(LOMMN,N)*FAC(LO)/(FAC(M)*FAC(LOMMN)) END;
          END;
  LAB3: WRITE "(",LO,") ",KAPPA(LO);
   X4:=KAPPA(3)/5;
```

⋮ （以下　繰り返し）

IV. 前の program の結果

```
COMMENT   THIS IS A REDUCE-2 FILE TO STORE THE RESULTS OF

          TAYLOR SERIES EXPANSION OF SECOND TYPE BY TYEXGO.RED;

KAPPA(0):=G$

KAPPA(1):=DG$
```

（続く）

```
KAPPA(2):=G*GY + D2G$

KAPPA(3):=3*G*DGY + DG*GY + D3G$

KAPPA(4):=3*G**2*GYY + G*GY**2 + 6*G*D2GY + 4*DG*DGY + D2G*GY + D4G$

KAPPA(5):=15*G**2*DGYY + 10*G*DG*GYY + 8*G*GY*DGY + 10*G*D3GY + DG*GY
**2 + 10*DG*D2GY + 5*D2G*DGY + D3G*GY + D5G$

KAPPA(6):=15*G**3*GYYY + 18*G**2*GY*GYY + 45*G**2*D2GYY + 60*G*DG*
DGYY + 15*G*D2G*GYY + G*GY**3 + 21*G*GY*D2GY + 18*G*DGY**2 + 15*G*
D4GY + 10*DG**2*GYY + 10*DG*GY*DGY + 20*DG*D3GY + D2G*GY**2 + 15*D2G*
D2GY + 6*D3G*DGY + D4G*GY + D6G$

KAPPA(7):=105*G**3*DGYYY + 105*G**2*DG*GYYY + 120*G**2*GY*DGYY + 84*G
**2*DGY*GYY + 105*G**2*D3GYY + 66*G*DG*GY*GYY + 210*G*DG*D2GYY + 105*
G*D2G*DGYY + 21*G*D3G*GYY + 15*G*GY**2*DGY + 45*G*GY*D3GY + 105*G*DGY
*D2GY + 21*G*D5GY + 70*DG**2*DGYY + 35*DG*D2G*GYY + DG*GY**3 + 31*DG*
GY*D2GY + 28*DG*DGY**2 + 35*DG*D4GY + 12*D2G*GY*DGY + 35*D2G*D3GY +
D3G*GY**2 + 21*D3G*D2GY + 7*D4G*DGY + D5G*GY + D7G$

KAPPA(8):=105*G**4*GYYYY + 225*G**3*GY*GYYY + 84*G**3*GYY**2 + 420*G
**3*D2GYYY + 840*G**2*DG*DGYYY + 210*G**2*D2G*GYYY + 81*G**2*GY**2*
GYY + 465*G**2*GY*D2GYY + 624*G**2*DGY*DGYY + 252*G**2*D2GY*GYY + 210
*G**2*D4GYY + 280*G*DG**2*GYYY + 508*G*DG*GY*DGYY + 360*G*DG*DGY*GYY
 + 560*G*DG*D3GYY + 113*G*D2G*GY*GYY + 420*G*D2G*D2GYY + 168*G*D3G*
DGYY + 28*G*D4G*GYY + G*GY**4 + 49*G*GY**2*D2GY + 82*G*GY*DGY**2 + 85
*G*GY*D4GY + 248*G*DGY*D3GY + 168*G*D2GY**2 + 28*G*D6GY + 66*DG**2*GY
*GYY + 280*DG**2*D2GYY + 280*DG*D2G*DGYY + 56*DG*D3G*GYY + 18*DG*GY**
2*DGY + 76*DG*GY*D3GY + 112*DG*DGY*D2GY + 56*DG*D5GY + 35*D2G**2*GYY
 + D2G*GY**3 + 43*D2G*GY*D2GY + 40*D2G*DGY**2 + 70*D2G*D4GY + 14*D3G*
GY*DGY + 56*D3G*D3GY + D4G*GY**2 + 28*D4G*D2GY + 8*D5G*DGY + D6G*GY
 + D8G$
```

V. (3.3)によって K_{il} をもとめる.

```
COMMENT  THIS IS A REDUCE-2 FILE TO CALCULATE IMPLICIT (1,0)-STAGE
         FORMULA EMPLOYING THE BELL POLYNOMIALS;

ARRAY BELL(15,15); OFF ECHO;

BELL(1,1):=X1$

BELL(2,1):=X2$

BELL(2,2):=X1**2$

   :  (BELL多項式のinput)

ON ECHO;
ARRAY KAPPA(2,8), DGY(8,7), RHO(2), NU(2), TAU(2,2),
      CAPTAU(2,5), CAPSIG(2,4), CAPRHO(2,3), QQ(2,8);
OFF ECHO;

DGY(0,0):=G;
DGY(0,1):=GY;
DGY(0,2):=GYY;
DGY(0,3):=GYYY;
DGY(0,4):=GYYYY;

   :  (DGY(I,J)のassignment)

DGY(8,6):=D8GYYYYYY;
DGY(8,7):=D8GYYYYYYY;

RHO(1):=RHO1;
RHO(2):=RHO2;

TAU(1,1):=TAU11;
TAU(1,2):=TAU12;

```

(続く)

```
TAU(2,1):=TAU21;
TAU(2,2):=TAU22;

NU(1):=NU1;
NU(2):=NU2;

CAPTAU(1,0):=CAPTAU10;
CAPTAU(2,0):=CAPTAU20;
```

:

（各arrayへのassignment）

·

```
COMMENT   INTEGER PROCEDURE FOR FACTORIAL OF AN INTEGER N;
  INTEGER PROCEDURE FAC(N);
        BEGIN INTEGER M;
                M:=1$
           L1:  IF N=0 THEN RETURN M;
                M:=M*N$
                N:=N-1$
                GO TO L1
        END;

FACTOR G,DG,D2G,D3G,D4G,D5G,D6G,D7G,D8G,GY,DGY,D2GY,D3GY,D4GY,D5GY,D6GY,D7GY,
      GYY,DGYY,D2GYY,D3GYY,D4GYY,D5GYY,D6GYY,GYYY,DGYYY,D2GYYY,D3GYYY,D4GYYY,D5GYYY,
      GYYYY,DGYYYY,D2GYYYY,D3GYYYY,D4GYYYY;

IMAX:=2;

FOR I:=1:IMAX DO KAPPA(I,0):=DGY(0,0);

L0:=1;
 INTEGER LOM1,M,N,LOMMN,MINI;
 LOM1:=L0-1;
 FOR I:=1:IMAX DO QQ(I,L0):=L0*(FOR J:=1:IMAX SUM(TAU(I,J)*KAPPA(J,LOM1)));
 I:=1;
  X1:=QQ(I,1);
  KAPPA(I,L0):=RHO(I)**L0*DGY(L0,0);
  IF L0<=1 THEN GO TO LAB11;
  FOR M:=1:LOM1 DO
    BEGIN MINI:=L0-M;
          IF M<=MINI THEN MINI:=M;
          FOR N:=1:MINI DO
            BEGIN LOMMN:=L0-M-N;
                  KAPPA(I,L0):=KAPPA(I,L0)+
                  RHO(I)**LOMMN*BELL(M,N)*DGY(LOMMN,N)*FAC(L0)/(FAC(M)*FAC(LOMMN))
            END;
    END;
LAB11: WRITE "(",I,",",L0,") ",KAPPA(I,L0);

 I:=2;
  X1:=QQ(I,1);
  KAPPA(I,L0):=RHO(I)**L0*DGY(L0,0);
  IF L0<=1 THEN GO TO LAB12;
  FOR M:=1:LOM1 DO
    BEGIN MINI:=L0-M;
          IF M<=MINI THEN MINI:=M;
          FOR N:=1:MINI DO
            BEGIN LOMMN:=L0-M-N;
                  KAPPA(I,L0):=KAPPA(I,L0)+
                  RHO(I)**LOMMN*BELL(M,N)*DGY(LOMMN,N)*FAC(L0)/(FAC(M)*FAC(LOMMN))
            END;
    END;
LAB12: WRITE "(",I,",",L0,") ",KAPPA(I,L0);

L0:=2;
 INTEGER LOM1,M,N,LOMMN,MINI;
 LOM1:=L0-1;
 FOR I:=1:IMAX DO QQ(I,L0):=L0*(FOR J:=1:IMAX SUM(TAU(I,J)*KAPPA(J,LOM1)));
 I:=1;
  X1:=QQ(I,1); X2:=QQ(I,2);
  KAPPA(I,L0):=RHO(I)**L0*DGY(L0,0);
```

（以下繰り返し）